

This is the author's copy of the publication as archived with the DLR's electronic library at <http://elib.dlr.de>. Please consult the original publication for citation.

Mobile manipulation for planetary exploration

P. Lehner; S. Brunner; A. Dömel; H. Gmeiner; S. Riedel; B. Vodermayr; A. Wedler

Keywords: aerospace robotics; manipulators; mobile robots; path planning; planetary rovers; telerobotics; complex manipulation tasks; arbitrary tool handling; moon-analogue demonstration mission; mobile manipulation; planetary exploration; robotic systems; foreign planets; teleoperation; autonomous rover; versatile constraint motion planner; remote control; autonomous task control; Light Weight Rover Unit; Task analysis; Payloads; Manipulators; Software; Computer architecture; Planets

Copyright Notice

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Citation Notice

```
@INPROCEEDINGS{8396726,  
  author={P. Lehner and S. Brunner and A. Dömel and H. Gmeiner and S. Riedel and B. Vodermayr and A. Wedler},  
  booktitle={2018 IEEE Aerospace Conference},  
  title={Mobile manipulation for planetary exploration},  
  year={2018},  
  volume={},  
  number={},  
  pages={1-11},  
  keywords={aerospace robotics; manipulators; mobile robots; path planning; planetary rovers; telerobotics; complex manipulation tasks; arbitrary tool handling; moon-analogue demonstration mission; mobile manipulation; planetary exploration; robotic systems; foreign planets; teleoperation; autonomous rover; versatile constraint motion planner; remote control; autonomous task control; Light Weight Rover Unit; Task analysis; Payloads; Manipulators; Software; Computer architecture; Planets},  
  doi={10.1109/AERO.2018.8396726},  
  ISSN={},  
  month={March},}
```

Mobile Manipulation for Planetary Exploration

Peter Lehner*, Sebastian Brunner*, Andreas Dömel, Heinrich Gmeiner,
Sebastian Riedel, Bernhard Vodermayr, Armin Wedler

German Aerospace Center (DLR)

Institute of Robotics and Mechatronics

Muenchener Str. 20, 82234 Wessling, Germany

{Peter.Lehner, Sebastian.Brunner, Andreas.Doemel, Heinrich.Gmeiner,
Sebastian.Riedel, Bernhard.Vodermayr, Armin.Wedler}@dlr.de

*Both authors contributed equally to this work.

Abstract—Robotic systems map unknown terrain and collect scientific relevant data of foreign planets. Currently, pilots from Earth steer these rovers on Moon and Mars surfaces via teleoperation. However, remote control suffers from a high delay of the long distance communication which leads to a reduction of the time the rover can spent gathering scientific data. We propose a system architecture for an autonomous rover for planetary exploration. The architecture is centered around a flexible, scalable world model to record and represent the environment of the robot. An autonomous task control framework and a versatile constraint motion planner use the live information from the world model to steer the rover through complex manipulation tasks. Furthermore, we present the enhancement of our Light Weight Rover Unit (LRU) with an innovative docking interface for arbitrary tool handling. We showcase the effectiveness of our approach at the moon-analogue demonstration mission of the ROBEX project on Mt. Etna, Sicily. We show in two experiments that the robot is capable of autonomously deploying scientific instruments and collecting soil samples from the volcano’s surface.

I. INTRODUCTION

Planetary exploration aims at discovering scientific insights about other planets. Scientists want to explain the planet’s composition and creation, find new elements, and ultimately discover new life on a remote planet. All these discoveries require data. Scientists want to analyze images of the planet surface, measure seismic activities and take samples of the planet’s soil. One solution is to send a robotic system to the remote planet in order to gather this data. Employing an autonomous rover has one main advantage: Data can be gathered faster than by human teleoperation as it is not hindered by the delay of a remote distance communication.

Nevertheless designing an autonomous mobile manipulation approach for planetary exploration poses many challenges. One issue is the mass of the robotic system. While the rover needs to be relatively lightweight, it needs to manipulate objects which are relatively heavy. Another issue is the communication delay. As live monitoring is limited, the robot must feature a high degree of autonomy and robustness. Finally many challenges of autonomous manipulation apply: Complex system architecture with many sensors and actors, partial information of the robots surroundings, and location uncertainty of the robot and the manipulated objects.

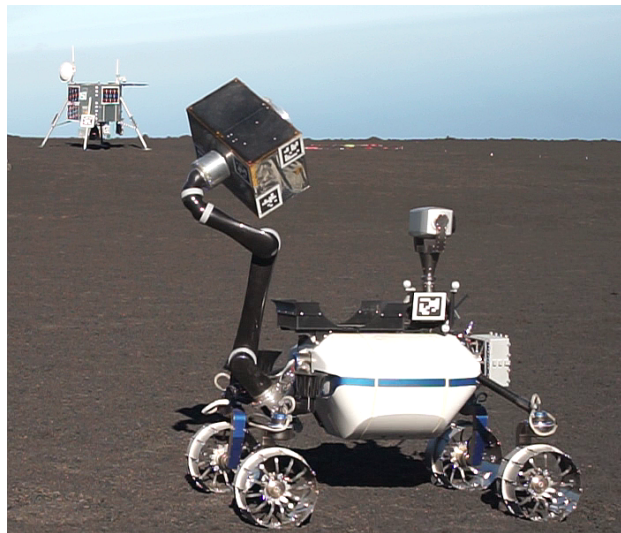


Fig. 1. The Light Weight Rover Unit lifts the scientific instrument from the ground and places the seismometer into the payload carrier on the robot’s back.

We present an integrated approach for mobile manipulation, which tackles the key challenges of planetary exploration. We feature a fully autonomous rover prototype with stereo cameras, a lightweight manipulator and an innovative docking interface. The rover is steered by our autonomous task control software with an architecture design centered around a flexible world model. The robot’s environment information is shared in a graph database between the core software components: A constrained motion planner, an autonomous navigation and exploration system, and object detection modules for lander and measurement instrument localization.

We showcase our approach and its robustness in two planetary exploration scenarios on a moon-analogue site on Mt. Etna, Sicily. In the first experiment, the rover delivers a seismographic instrument in a payload box from a landing unit to a chosen location. The rover autonomously deploys the instrument, which includes leveling of the ground surface and creating an impulse for a test measurement. In the second

experiment, the rover takes a payload box with storage slots for soil samples from a landing unit and deploys it at a target location. The robot then docks to a shovel, acquires a sample, inserts it into the payload box, and recovers the payload box from the ground. Both missions pose significant challenges to the rover’s autonomy but were completed successfully.

II. RELATED WORK

Real space environments like on the Moon or Mars poses many challenges to both the hardware and software of a rover. The most famous examples of rovers ever built are *Sojourner* [1], *Lunokhod* [2], *Spirit*, *Curiosity* and *Opportunity* [3]. All of them were deployed on the Moon or Mars and covered distances from 100 meters to more than 40 kilometers [4]. Unfortunately, they all lack an autonomous behavior as all their actions are remotely controlled [2, 3, 5]. Thus, new systems evolve, which attempt to complete missions fully autonomous, like the *Light Weight Rover Unit* (LRU) rover [6].

Robot autonomy requires flexible system and software architectures to divide the complexity of the overall task. The *Clarity* architecture [7] was proposed at the beginning of this century, but was not implemented on a real Moon or Mars mission. Further software architectures proposed by NASA are [8] and [5], which generate action sequences automatically but are double-checked by humans each time before they are sent to the Mars rovers. Architectures for fully autonomous systems are proposed and tested by Schuster et al. [6], Eich et al. [9] and Schneider et al. [10].

Autonomy requires frameworks for task programming, behavior definition, and mission control. We designed and implemented our own framework as all existing software did not fit our needs [11]. Unfortunately, the development and support for many software packages already discontinued [12, 13], although the general concepts behind were quite promising. Other frameworks did not ship a graphical editor for robotic behavior, which is, in our opinion, vital to cover very complex scenarios [14, 15]. Thus, we developed our own solution for creating complex, robotic tasks which is called *RMC Advanced Flow Control* (RAFCON) [11]. The best alternative to our framework is *FlexBE*, which supports many of the features of RAFCON but lacks semantic state annotation, generation of meaningful task metrics during state machine creation or execution, and post-mortem analysis [16].

The robotics open source community lacks software packages for semantic knowledge representation. There are two *Robot Operating System* (ROS) packages [17, 18], which are both discontinued. Apart from ROS there are few other world model frameworks available, and some of them represent their knowledge in a scene graph-like structure [19, 18, 20, 21]. In contrast, we integrate a graph database for the storage of highly connected heterogeneous data and designed a flexible interface for distributed remote access to the world model, which is presented in Section III-B.

Planning motions for manipulators is a wide field in the robotics community. Sampling based planners like the

Rapidly Exploring Random Tree (RRT) and the *Probabilistic Roadmap Method* (PRM) have emerged as the most widely used planning approaches due to their ability to explore the vast solution spaces of the manipulators configuration space efficiently and with probabilistic completeness [22, 23]. Our motion planner works in concept similar to the *Constrained Bi-RRT* (CBiRRT2), which extends the RRT algorithm with the concept of task constraints [24]. Our approach exceeds the CBiRRT2 by its tight integration in the system architecture and its additional constraints for planetary exploration, for example gravitational constraints. Additionally our approach has the ability to learn from previous planning queries which is presented in detail in [25].

Docking and interfacing systems targeting deep space and planetary missions differ greatly in their requirements. They are usually adapted to narrowly defined tasks, lowering risks for the mission and costs, but leaving no options for flexibility. Common functionalities of these systems are that they enable connection of mechanical loads, transfer of electrical power and data as well as establishing thermal distribution by liquid exchange. Despite the synergies between these fields of interest, several approaches by renowned agencies lead to the implementation of individual, proprietary designs. The docking of mechanical loads can be solved through exposed guide elements [26, 27]. As a result, the active interface part must be capable of generating forces towards the passive coupling partner, which are high enough to accomplish mating within the correct orientation followed by establishing the connection for the target application. High forces have to be controlled and monitored carefully as docking actions can bear the risk of collision with other parts of the system. As systems often employ space grade components of the shelf with rectangular shape, the mating and connecting requires a high precision to lower the risk of system damage and mission failure.

One way to classify the process of robotic manipulation and the necessary components involved is by the weight of the handled payload. Solutions comprising a gripper that provides functionality similar to that of a human hand are in general more appropriate for the manipulation of light weight objects. Compared to this the handling of heavier payloads requires an interface comparable to an industrial tool changer. However, for the application of such a system the manipulator platform has to provide precise positioning and a static model of the working environment has to be established for the time of operation. Dynamic changes of the environment lead to a time-consuming update of the model and even technical components, possibly rendering the system unable to operate correctly. In consequence, standard industrial tool changers contradict to systems required for explorative tasks performed by mobile robots in an unknown environment. In such cases the world model is changing dynamically, thus systems involved in docking or undocking maneuvers have to provide a capability for larger tolerances.

III. MOBILE MANIPULATION FOR PLANETARY EXPLORATION

Planetary exploration missions contain key milestones which define the mission success or failure. One example of such a milestone in our experiments is when the rover docks to the payload element on the lander. To reach this milestone autonomously the rover solves a whole sequence of tasks: The rover must locate the landing unit, drive to the correct docking location, fine position the rover body, execute a manipulator motion to position the docking interface, and dock rigidly to the payload element. Every step must position the rover, the manipulator, or the docking interface within a certain distance of the payload for the next step to work. As this describes a long chain of dependencies the main difficulty is not each individual step (nevertheless hard by themselves) but the robustness of the integrated system.

To achieve a robustness level to execute a whole planetary exploration mission, we designed an architecture which is focused on integrating all components, as shown in Fig. 2. The centerpiece of the architecture is a world model which holds all the information the rover knows about itself, its environment, as well as the mission status. This information is shared by the main software components: The control flow RAFCON, the motion planner for the manipulator, the navigation and exploration modules, and the object detection and localization components. To control the access to the world model we designed special interfaces, which provide different views onto the world model. For example, the motion planner gathers its required information through an interface which only exposes the rigid body kinematics of the current scene. By having all information in one central representation with controlled access, we can ensure that each component always receives the current data and does not access data which is irrelevant for its domain.

Next to this central world model the main components for mobile manipulation in our system are the task control framework, the navigation and exploration module, the manipulator motion planner, and the docking interface. To build on our previous work we only briefly present the individual components and focus on the novel contributions centered around the world model. As the navigation and exploration component remained similar in concept please refer to our previous publication for details [6].

A. Autonomous Task Control

To program the autonomous behavior of the LRU rover and to monitor the progress of the mission execution we employed our flow control software RAFCON [11]. Compared to the state machines programmed for our previous experiments (see [6]) the complexity of the autonomous behavior for our scenarios in the project *Robotic Exploration of Extreme Environments* (ROBEX) rose even more. For the seismic measurements (see Section IV-A) we created state machines of more than 1400 states and 1900 transitions of a maximum hierarchy level of eight.

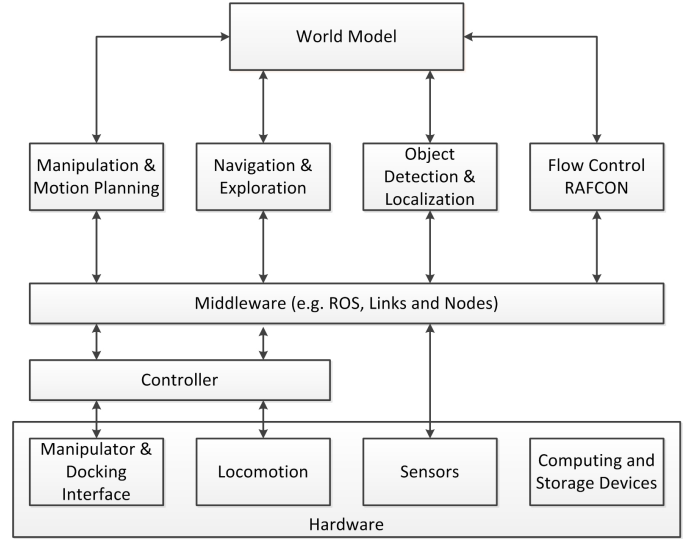


Fig. 2. The abstract system architecture of the Light Weight Rover Unit, depicting the main components.

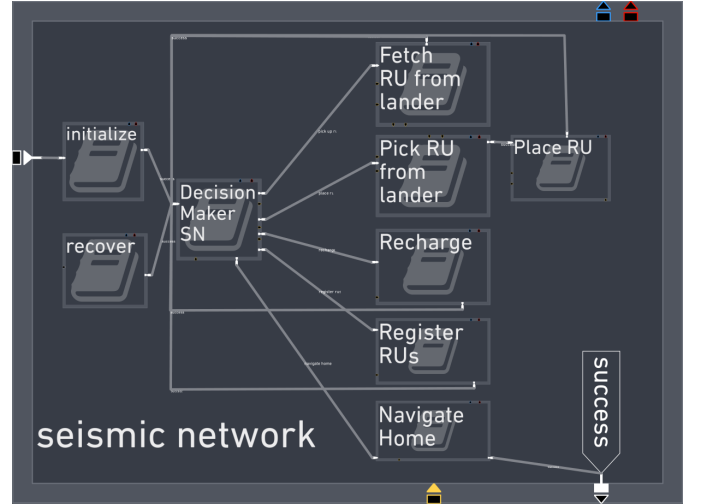


Fig. 3. The RAFCON state machine for the seismic measurement experiment described in Section IV-A.

As described in [6], we use a main decision maker for the robot to decide with subtask to execute next based on the knowledge of the robot's environment (i.e. the world model). Moreover, we employ local decision makers to decide upon the next actions based on different sensor inputs or failure events. Error recovery procedures for many failure types can be added easily as RAFCON offers powerful support for error handling in its very core design.

We extended the robot behavior to cover different autonomy levels depending on the current testing situation. In general, the main mission commands are issued by the sequencer on the landing unit, so the rover has to synchronize with the lander. Furthermore the rover has to communicate with the seismic measurement units (or "remote units") to retrieve e.g. the gravity vector of the current deployment pose. As usual

for real outdoor tests, both the communication to the lander and to the remote units happens to fail. In these situations the robot needs to choose the best action path to continue the mission as good as possible.

RAFCON furthermore enables us to test the whole action sequences remotely from a control computer in the base station before it is deployed onto the robot for fully autonomous behavior. In practice, this development procedure saves a lot of time, as a frequent deploying procedure over a bandwidth limited, high delayed communication would block the robot for long durations.

B. World Model

1) *Basic Concept:* In the last years the development of robotic systems has benefited immensely from the design concept of a fine grained modular approach. One major goal is to decouple all components. So why do we come up with a robot world model as a central component? The consideration is that most components have an internal world model. The world models of some components are a complex, geometric representation, e.g. for path planning modules; others are only a few parameters identifying the systems properties for control. However, since almost all modules are related to the real world, the components indirectly depend on each other. For example if an object is grasped the load data changes, additional constraints arise for the planner, the field of view of the cameras change, etc. Instead of keeping all models individually updated our approach has one model of the world. Changes of the real world are tracked and modeled in that component. All other components can extract their world representations from this world model: Ensuring synchronization.

Since computational resources on the system are limited, we use an abstract representation of the world, instead of simulating the environment and physical effects in detail. The basic consideration is that the most important relation between objects is the geometric relation. Hence our world model is based on geometric transformations connecting different objects. Since our use cases are usually in quasi-static environments most of these relations are constant. Instead of simulating physical laws our assumption is: Every transformation is constant until we get new information. We decided to use a tree structure due to the fact that most of the knowledge we have about the environment are relative relations. For example we know the static relation between lander and docking port, but not the pose of the docking port in the map frame. Therefore every object has exactly one parent object with a defined transformation. Changing this relation leads implicitly to changed poses of all child objects regarding the map frame. Besides physical objects our world model has different object types, like e.g. frames, markers and grasps representing additional knowledge in the model (see Figure 4).

2) *Application:* For manipulation tasks the robot has to change its environment. These changes of the environment can be modeled explicitly. Since our world model representation of the robot is very simple, consisting of two objects, only few operations and their effects are possible during execution:

- Move manipulator: When the robot moves its manipulator the transformation between robot base and flange has to be updated.
- Move platform: When the robot's platform is moved, the transformation between the scene root and robot base has to be updated.
- Pick up object: When an object is grasped, the parent of the object has to be changed to the robot flange in the world model. The transformation to the robot flange is given by the applied grasp.
- Place object: When an object is placed onto an object in the scene, the parent of the object changes from the robot flange to the object on which the object was placed.

To estimate a pose of an object a local reference frame is needed. Knowing the world pose of the reference frame allows computing the world pose of the object. Often the same reference object or frame is used for many objects. Given the case that the measurement of the reference object to the world frame was improved, all referenced objects have to be moved to benefit from that information.

3) *Implementation:* The proposed world model is implemented using the graph database *Neo4j*¹ as backend. Neo4j represents data using a property-graph model, meaning the database is a graph composed of nodes and interconnected by edges, both of which can store arbitrary properties in a key-value fashion. For our world model, we directly represent objects in the world model as nodes in the database graph and correspondingly edges in the graph build up the world model tree structure. The relative position and orientation between two objects are stored as properties of the connecting edge. We define and ensure a common type-system for objects in the world, by defining a type-hierarchy with precisely listed required and optional properties for each object type (e.g. every *PhysicalObject* has to have a mass; every *Grasp* a width and force). This is done via a so called *object-graph-mapping* (OGM) using the *neomodel* library².

A database as backend in general provides useful features to keep the world model consistent at any time, e.g. synchronized multi-client read-/write-access and transaction support for batching multiple modifications into an atomic operation. In particular, we use transactions with pre-/post-transaction sanity checks to allow complex yet safe operations on the world model. In case of a software error or violated sanity checks (e.g. a world model object suddenly has two parents), the world model is automatically rolled back to the consistent state before this modification. As a graph database, Neo4j offers a pattern-related query language called *CYPHER*, which makes it very easy to query the world model for e.g. "all rigid bodies for which a grasp is defined."

We use YAML-files as a human read-/writeable way to specify the initial world model, but also complex object templates (e.g. complete subgraphs of *PhysicalObjects* together with attached *Grasps* and *Markers*, etc.). To add such tem-

¹<https://neo4j.com/>

²<https://github.com/robinedwards/neomodel>

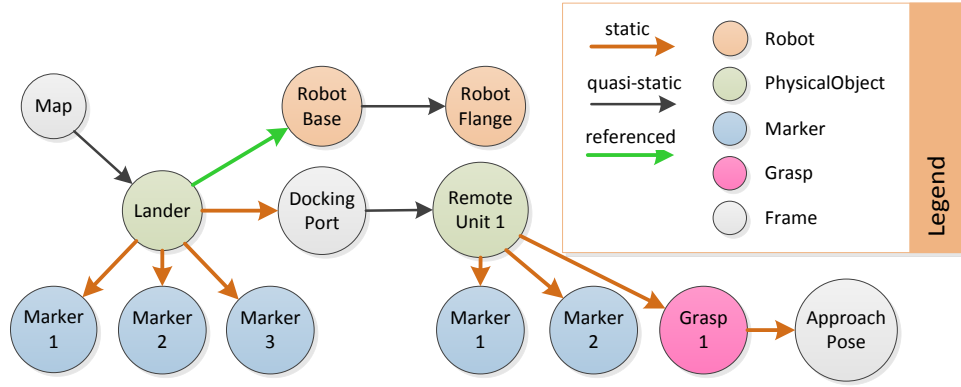


Fig. 4. Schematic of a part of the world model after the robot referenced itself to the *Lander*, shortly before picking up the *Remote Unit*.

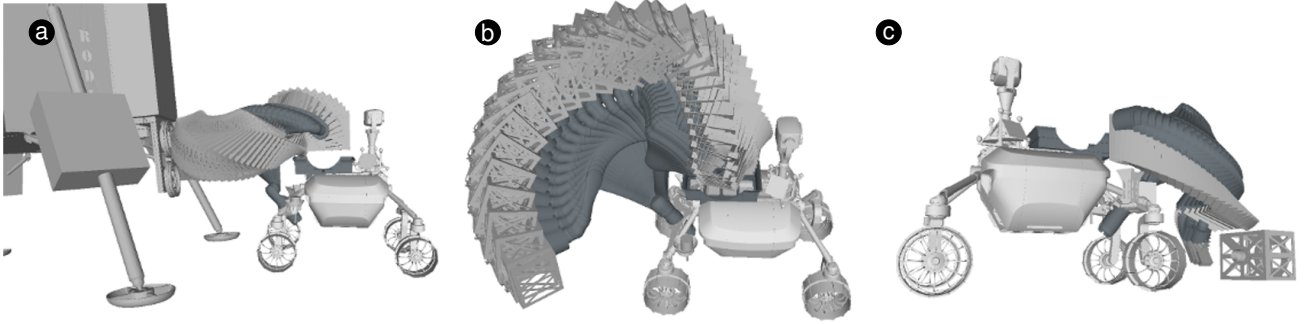


Fig. 5. Different solution paths found by the manipulator motion planner shown as discretized configurations of the arm. a) shows the manipulator motion of moving the payload from the lander onto the rover, b) shows the motion of picking up the payload from the ground, and c) shows the motion of approaching the ground with a shovel.

plates into the world model, we developed a domain-specific module and API for higher level operations based on the raw CYPHER- and OGM-access to the database. Other more complex operations include querying the relative transformation between arbitrary objects in the graph or maintaining unique labels for nodes (e.g. exactly one node might hold a label *CURRENT_PLANNING_SCENE*).

This API is the basis for several adapter modules which make operations accessible via ROS services. New functionality or composite operations can be added in a modular way by writing additional adapters as necessary. Two examples would be 1) the GeometricScene-Adapter, which provides services to get a reduced object tree with everything relevant for building a geometric scene representation for motion planning, and 2) a ROS-Tf-Adapter, which takes care of periodically publishing the world as ROS-Tf-tree as well as updating externally published transformations (for example *robot_flange* \rightarrow *robot_base*) in the world model.

C. Manipulator Motion Planning

For the exploration of a remote planet scientists want to deploy scientific instruments, take samples of the soil, and perform maintenance on the landing equipment. To equip a rover with a general purpose device for all of these tasks a

robotic manipulator can be employed. But if the rover is to autonomously use the manipulator, it must plan motions which satisfy the following constraints:

- The motion must be collision free, permitting only desired contacts with the environment.
- The motion must respect the manipulators kinematic and dynamic constraints: The manipulators overall structure and the joint and torque limits.
- The motion must fulfill the required task: For example connect the docking interface to the payload box.

Planning these motions is particular difficult as the search space, the configuration space, of the manipulator is vast. To efficiently search the space of possible solutions we employ a sampling based planner, as previously presented [6]. By using a derivation of the *Rapidly Exploring Randomized Tree* (RRT) we are able to compute a motion for the manipulator which satisfies all the mentioned constraints [22]. Figure 5 shows example solutions for tasks solved during the experiments.

Building on the previous work we have included two major new concepts in the manipulator motion planning:

- Synchronization of the manipulation planner with the world representation.
- Torque constraints to manage the high payload of the seismic instruments.

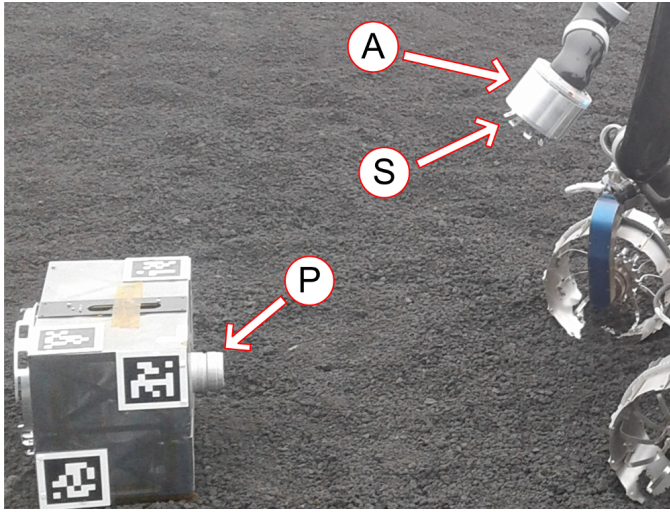


Fig. 6. The payload box with passive coupling partner “P” placed on the ground (left side). The active coupling partner “A” comprising the metal spring grasping elements “S” (right side) while opening the capturing zone.

The synchronization of the manipulation motion planner is a central component of the overall manipulation pipeline. To safely plan and execute motions of the manipulator, the motion planner always requires the latest model of the rover’s state and the environment. This information is always present in the world model as described in Section III-B. To extract the geometric information, we designed a specialized interface: The rigid body interface extracts all geometric and shape information from the graph database and offers this information to the motion planner for a specific scene. For example before the rover manipulates an object close to the lander, it detects the precise location of the lander and the surrounding objects. This information is inserted into the world model by the perception processes of the rover. Once the rover plans a motion for its manipulator, it extracts the shape and geometric relations of the lander and the surrounding objects to generate a collision free motion.

Respecting the torque limits of the manipulator is particular important in planetary exploration. On the one hand the robotic manipulator must be lightweight to limit the cost of sending it to the remote planet. On the other hand, the manipulator must lift relatively heavy payload elements to deploy the scientific instruments. This leads to the fact that the manipulator cannot hold all payload elements at every configuration. For example, if the manipulator is fully stretched, the torques at joint two exceed the joint limits by far. To avoid such configurations, we include the torque limits when planning motions. At each configuration the planner computes the expected gravitational torques at the arm joints and selects a path which satisfies the torque limits of the manipulator.

D. Envicon Docking Interface

To avoid high precision positioning of the docking interface at the beginning of the docking process, we developed a

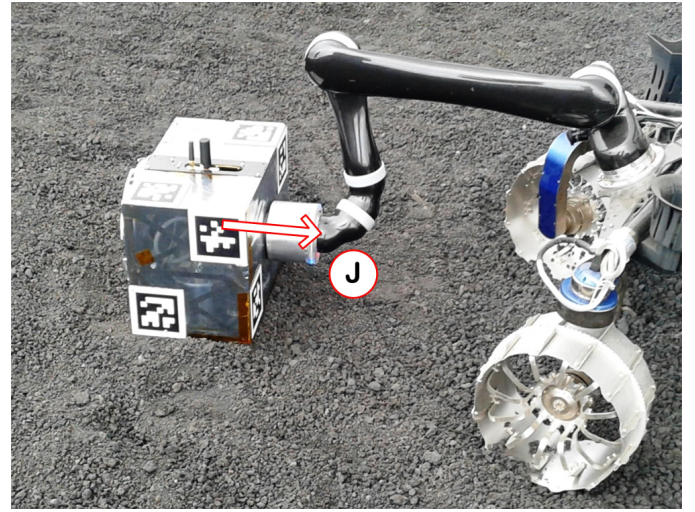


Fig. 7. The mounting point of the docking interface’s active part is shifted behind the robotic arm’s TCP towards the last joint “J” in order to keep the distance between the payloads center of gravity and the joint as short as possible.

concept that uses a rotationally symmetric geometry for the docking core including the connection of mechanical loads, electrical power, data transfer and fluid transfer. Furthermore, the concept foresees the principle of retraction accomplished by the active part of the docking interface, providing a zone of higher tolerance to initiate the docking process instead of requiring exact prepositioning by the robot or vehicle as described in [28], [29], [30] and [31]. The presented concept is capable of increasing the misalignment tolerance, thus leading to a more robust docking process especially on mobile robots in rough and undefined environments. Furthermore, the concept can lower the overall system weight, as the forces required during the docking process do not have to be produced by the manipulation platform. In fact, the forces provided by the interface during the docking procedure exceed those of the manipulator platform. Despite the lack of guide rails in the docking core, the traction and friction created by the interface can withstand maximum torques of the robotic arm. Figure 6 shows an overview of the docking interface in its validation environment.

The docking interface consists of an active and a passive coupling partner. As proposed by the novel concept, the latter is a rotationally symmetric cylinder with a defined, partially conic shape. The active coupling partner is basically a cylindrical structure with an outer diameter of 102mm, an inner diameter of 65mm and a length of 79mm. The cylindrical structure incorporates the mechanical capturing mechanism and the system controller unit along with redundant sensors and power supplies. The system controller runs a set of software processes that serve for inter-system communication, safety mechanisms and the control of the docking process. The design was adapted to the robotic arm used during the mission, leading to the interfaces overall weight of 390g while being capable of safely docking and manipulating payloads of more

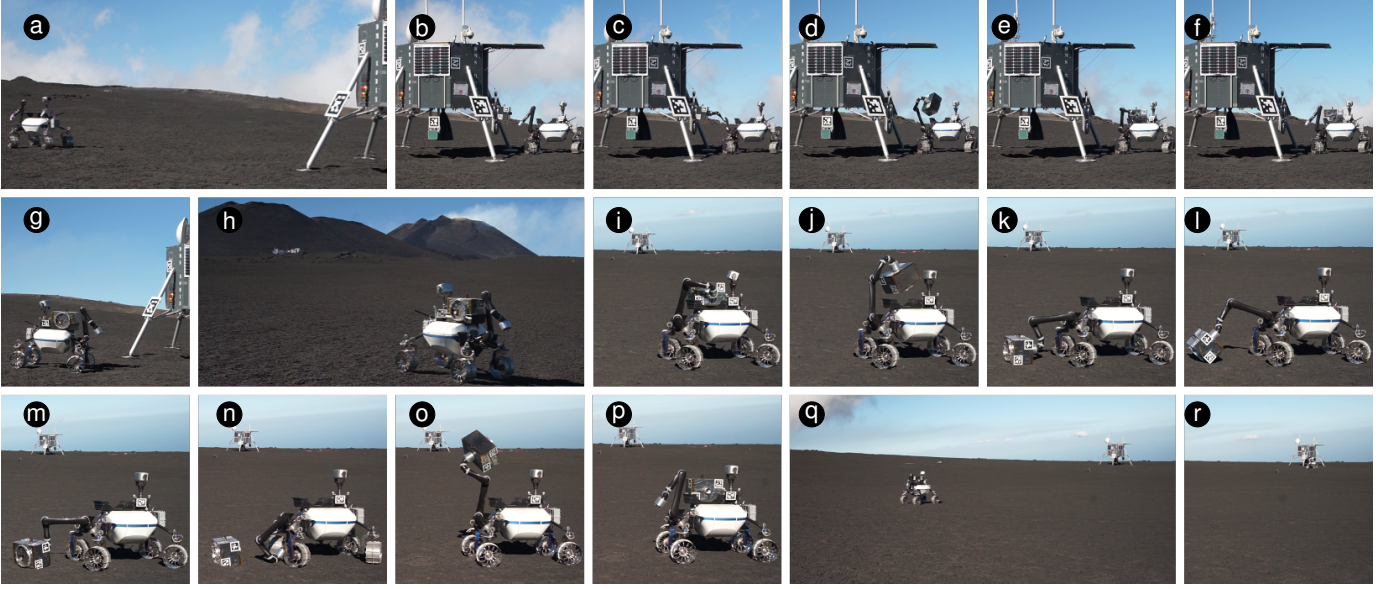


Fig. 9. Main scenes from the seismic measurement mission. The LRU rover picks the seismograph from the lander (a-f), drives to the deploy location (g-i), deploys the instrument (j-n), and returns to the lander (o-r).

B. Sample Return

The target of the second experiment is to collect a soil sample from a target location. The rover approaches the region of interest and places the probe container on the ground. After grasping the shovel from a special holder element on the robot's side, it shovels a soil probe into the probe container. Subsequently, the rover puts the sample box back onto the storage of the rover's back and returns the sample to the base station.

Next to planning collision free movements in a reasonable amount of time, another main challenge is to consider special constraints while manipulating the target objects. Specifically, while placing the soil sample into the sample container, the motion planner must ensure that the manipulator does not spill any of the collected soil. The same consideration has to be taken into account for the probe container after filling it with the soil. Arbitrary motions of the box could spill the stored sample. These orientation constraints for the docked objects must be considered by the motion planner.

Overall the rover was able to successfully complete the sample experiment fully autonomously in about ten minutes. Figure 10 shows the main scenes from the experiment. At the beginning the rover drove to the sample location (a) and deployed the probe container by docking to the container and afterwards planning and execution a motion to place it onto the ground (b-d). After subsequently docking to the shovel with the manipulator, the rover used the shovel to collect a probe of the ground (e) and inserted the probe into one of the slots of the probe container (f). Once the manipulator had stored the shovel again, the rover lifted the probe container without spilling the probe (g-i).

V. DISCUSSION

The successful execution of both experiments shows that the LRU rover can complete complex mobile manipulation tasks for planetary exploration. The system's main feature is that it can complete the tasks autonomously, robustly, and within a relatively short time. Both the autonomy and the robustness stem from the presented system architecture with the central world model. By gathering all information centralized and distributing the data over specialized interfaces to the individual components, each component is informed of the current world state, but only accesses the information relevant for its data.

Compared to the mars rovers of NASA [3] our rover is able to act more autonomously. In general, at NASA the autonomous behavior for a robot is generated on earth e.g. via MAPGEN [5]. Thereafter a human operator checks the action sequence, which is subsequently sent to the rover on Mars. This cycle is likely to consume a lot of time, as the communication delay between Earth and Mars can be up to 24 minutes. As described in Section III, our rover is capable of deciding upon the next action sequence by evaluating the autonomous collected data of the environment.

The task execution software RAFCON provided the ability to capture the complex mission procedure in human understandable state machines. The hierarchical concept of the state machines allowed us to encapsulate the mission parts into clear segments and intuitively compose these segments into the overall mission. The intuitive visualization of the control flow during programming as well as testing allowed for swift debugging and monitoring of the rover's process in the mission.

Our integrated manipulator motion planner was able to compute all paths necessary for the individual manipulation

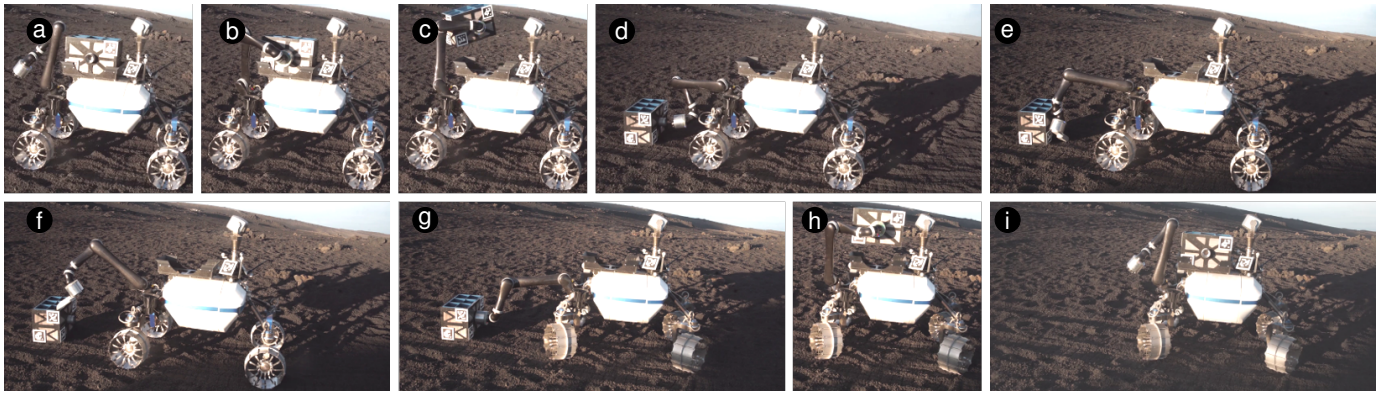


Fig. 10. Main scenes from the sample return mission. The LRU rover autonomously places the probe container (a-d), inserts a sand probe into the container with a shovel (e-f), and lifts the probe container again (g-i).

steps. During the sample experiment, the motion planner was able to plan constrained motions for inserting the extracted soil with the shovel into the probe container. Planning this motion was particular hard due to the narrow solution space within the vast search space. During the seismic measurement experiment, the motion planner was able to generate paths for picking the payload from the lander, placing the payload on the soil, executing a test impulse as well as picking the payload from the ground again. Handling the heavy payload required the planner to observe the torque constraints induced through gravity.

The envicon docking interface enabled the rover to rigidly connect to a versatile range of objects. In the probe experiment the rover was able to dock rigidly to the probe container as well as the shovel. Both connections withstood all torques induced to the manipulation steps especially during shoveling the ground. In the seismic measurement experiment, the docking interface withstood the torques induced through the heavy payload of the seismograph. Additionally, the docking interface did not succumb to the adverse conditions on Mt. Etna and was able to resist the fine lava dust as well as high wind speeds over a test campaign of several weeks.

Although our system can perform the majority of the tasks autonomously and in a robust manner, there are still many problems our rover cannot tackle. Especially, weather constraints complicate object detection and lead to lower object detection success rates or inaccurate pose estimations. Furthermore, heavy wind put additional constraints on the impedance controller for in-contact motions. In summary, there are many open challenges that have to be tackled to build a truly robust system able to cover a wide field of outdoor scenarios.

An open issue is also the verification of the system autonomy to not perform any actions which endanger the mission. Since the system consists of many distributed processes, an exhaustive formal verification of all individual processes and the overall integration would pose a momentous effort. Therefore we propose three aligned strategies to reach the necessary robustness: A verification of the autonomy on the

state machine level in RAFCON, which can be automatized and is thus feasible; an error recovery concept which leverages the hierarchical composition of the state machines as well as the centralized information of the world model; and extensive testing in mission analog test sites, e.g. on Mt. Etna.

VI. CONCLUSION

In this paper we proposed a robust and scalable system architecture for autonomous robots in the context of planetary exploration. The main features of this architecture consists of the online motion planning of highly constraint tasks and a powerful flow control framework closely linked to a world model capable of storing arbitrary information about the rover, its environment and gathered scientific data. We showcased our approach on several experiments in the context of the moon-analogue demonstration mission on Mt. Etna, Sicily. Ultimately, we could prove the robustness of our system in a rough terrain environment in the presence of harsh weather conditions including heavy wind and changing light and temperature conditions.

REFERENCES

- [1] R. Washington, K. Golden, J. Bresina, D. Smith, C. Anderson, and T. Smith, "Autonomous Rovers for Mars Exploration," in *IEEE Aerospace Conference*, vol. 1, Snowmass at Aspen, Colorado, 1999.
- [2] V. Gromov, A. Kemurdjian, A. Bogatchev, V. Koutcherenko, M. Malenkov, S. Matrossov, S. Vladykin, V. Petriga, and Y. Khakhanov, "Lunokhod 2 - A retrospective Glance after 30 Years," in *EGS - AGU - EUG Joint Assembly*, Apr. 2003.
- [3] J. P. Grotzinger *et al.*, "Mars Science Laboratory Mission and Science Investigation," *Space Sci. Rev.*, vol. 170, no. 1, pp. 5–56, 2012.
- [4] "Out-of-this-world records!" <https://www.jpl.nasa.gov/images/mer/2014-07-28/odometry140728.jpg>, accessed: 2017-08-07.
- [5] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J.-J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias, *et al.*, "Mapgen: mixed-initiative planning and scheduling

- for the mars exploration rover mission,” *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 8–12, 2004.
- [6] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Bttner, A. Dmel, W. Friedl, I. Grix, M. Hellerer, H. Hirschmüller, M. Kassecker, Z.-C. Marton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, “Towards autonomous planetary exploration: The lightweight rover unit (lru), its success in the spacebotcamp challenge, and beyond,” in *ICARSC - IEEE International Conference on Autonomous Robot Systems and Competitions*, 2016.
 - [7] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das, “The CLARAty architecture for robotic autonomy,” in *Aerospace Conference, 2001, IEEE Proceedings.*, vol. 1. IEEE, 2001, pp. 1–121.
 - [8] V. Verma, A. Jónsson, R. Simmons, T. Estlin, and R. Levinson, “Survey of command execution systems for NASA spacecraft and robots,” 2005.
 - [9] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, and J. Wollenberg, “Towards Coordinated Multirobot Missions for Lunar Sample Collection in an Unknown Environment,” *J. Field Robot.*, vol. 31, no. 1, 2014.
 - [10] F. E. Schneider, D. Wildermuth, and H.-L. Wolf, “EL-ROB and EURATHLON: Improving Search & Rescue Robotics through Real-World Robot Competitions,” in *International Workshop on Robot Motion and Control (RoMoCo)*. Poznan, Poland: IEEE, 2015, pp. 118–123.
 - [11] S. G. Brunner, F. Steinmetz, R. Belder, and A. Doemel, “RAFCON: A Graphical Tool for Engineering Complex, Robotic Tasks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7759506/>
 - [12] H. Nguyen, M. Ciocarlie, K. Hsiao, and C. C. Kemp, “Ros commander (rosco): Behavior creation for home robots,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 467–474.
 - [13] M. Loetzsch, M. Risler, and M. Juengel, “XABSL - A pragmatic approach to behavior engineering,” in *Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, Beijing, China, 2006, pp. 5124–5129.
 - [14] J. Bohren and S. Cousins, “The smach high-level executive [ros news],” *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 18–20, Dec 2010.
 - [15] M. Beetz, L. Mösenlechner, and M. Tenorth, “CRAM – A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18–22 2010, pp. 1012–1017.
 - [16] S. G. Brunner, P. Lehner, M. J. Schuster, S. Riedel, R. Belder, A. Wedler, D. Leidner, F. Stulp, and M. Beetz, “Design, Execution and Post-Mortem Analysis of Prolonged Autonomous Robot Operations,” in *submitted to IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018.
 - [17] “An object-based semantic world model,” <http://wiki.ros.org/worldmodel>, accessed: 2017-08-07.
 - [18] “Spatial world model for object tracking,” http://wiki.ros.org/spatial_world_model, accessed: 2017-08-07.
 - [19] S. Blumenthal, H. Bruyninckx, W. Nowak, and E. Prassler, “A scene graph based shared 3d world model for robotic applications,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 453–460.
 - [20] J. Elfving, S. van den Dries, M. Van De Molengraft, and M. Steinbuch, “Semantic world modeling using probabilistic multiple hypothesis anchoring,” *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, 2013.
 - [21] J. Mason and B. Marthi, “An object-based semantic world model for long-term change detection and semantic querying,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on.* IEEE, 2012, pp. 3851–3858.
 - [22] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2004.
 - [23] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
 - [24] D. Berenson, S. S. Srinivasa, and J. Kuffner, “Task space regions: A framework for pose-constrained manipulation planning,” *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.
 - [25] P. M. Lehner and A. Albu-Schäffer, “Repetition sampling for efficiently planning similar constrained manipulation tasks,” in *Proc. 2017 IEEE Int. Conf. Intelligent Robots and Systems*. IEEE, 2017, p. accepted.
 - [26] S.-I. Nishida and T. Yoshikawa, “Development of space robot end-effector for on-orbit assembly,” *Journal of the Japan Society for Aeronautical and Space Sciences*, vol. 53, no. 614, pp. 130–138, 2005.
 - [27] M. Nilsson, “Heavy-duty connectors for self-reconfiguring robots,” in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on.* IEEE, 2002.
 - [28] P. Roberts, “A novel passive robotic tool interface,” in *15th European Space Mechanisms & Tribology Symposium - ESMATS*, 2013.
 - [29] A. Dettman, Z. Wang, W. Wenzel, F. Cordes, and F. Kirchner, “Heterogeneous modules with a homogeneous electromechanical interface in multi-module systems for space exploration,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011.
 - [30] W.-M. Shen, R. Kovac, and M. Rubenstein, “Singo: A single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing,” in *Robotics and Automation, 2009. ICRA '09. IEEE Inter-*

- national Conference on.* IEEE, 2009.
- [31] R. Gelmi, A. Rusconi, J. Gonzalez Lodoso, P. Campo, R. Chomicz, and A. Schiele, “Design of a compact tool exchange device for space robotics applications,” in *Proceedings of the 9th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ASTRA*. ESTEC, 2006.